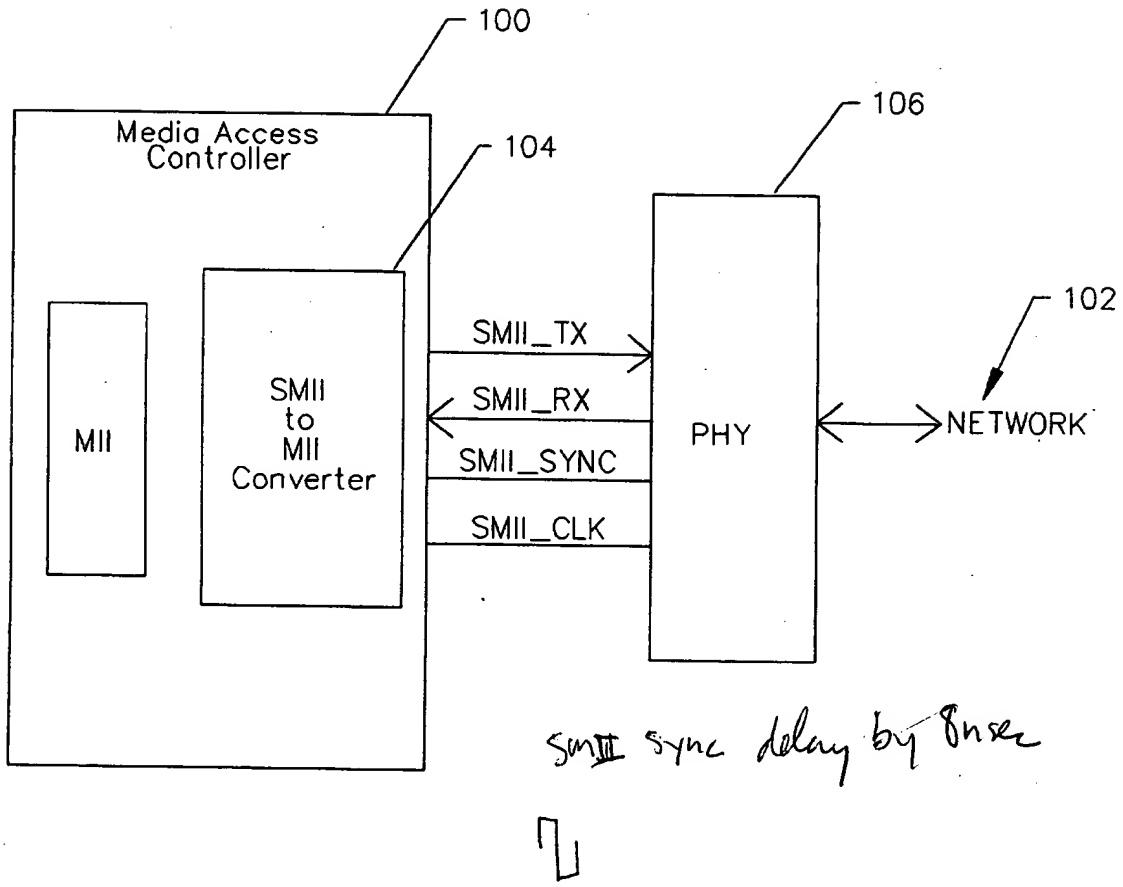
100
Media Access Controller
104
106
102
MII
SMII to MII Converter
SMII_TX
SMII_RX
SMII_SYNC
SMII_CLK
PHY
NETWORK
SMII sync delay by 8nsec

FIG.1

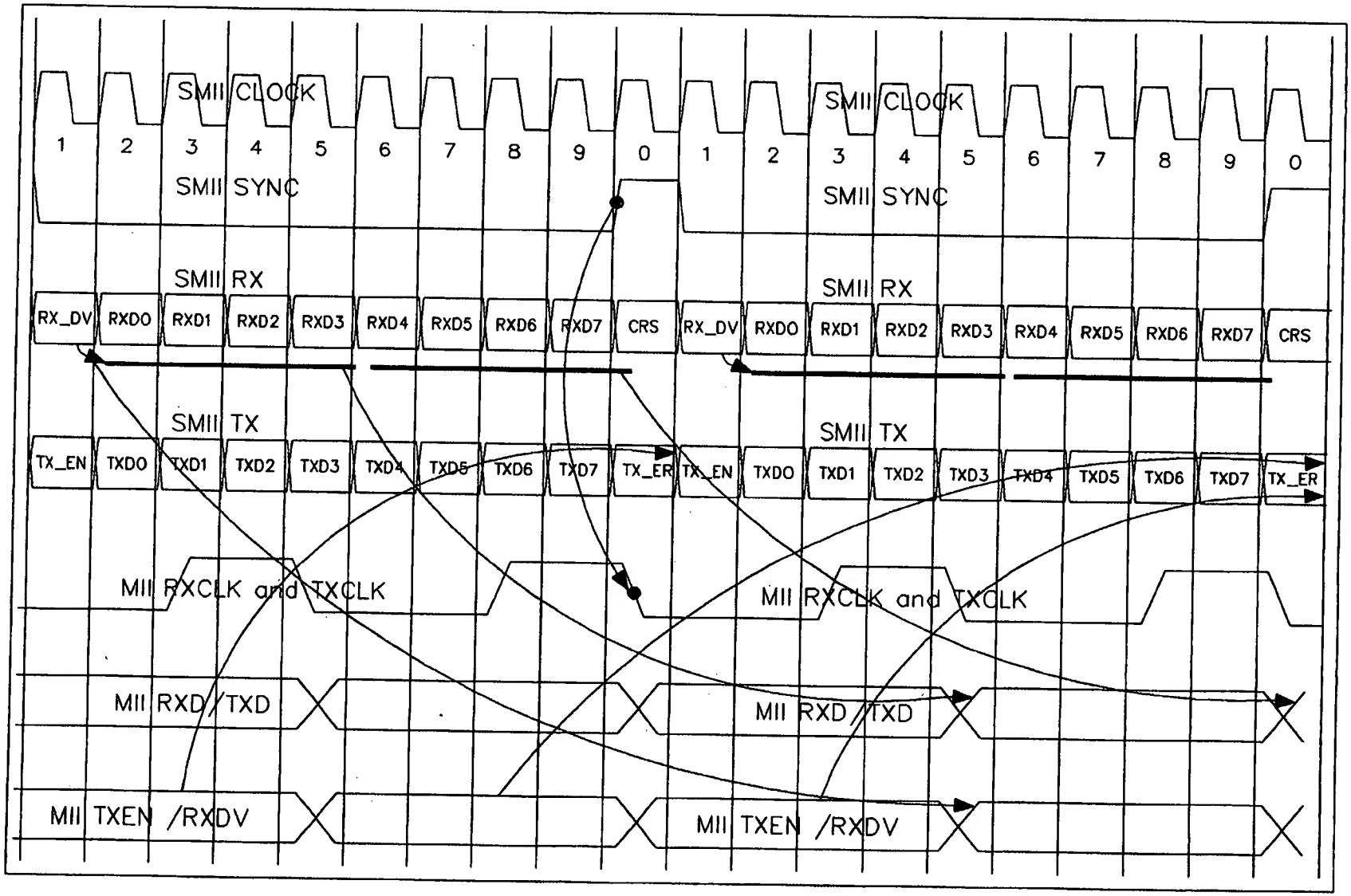
SMII CLOCK
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
SMII SYNC
SMII RX
RX_DV RXD0 RXD1 RXD2 RXD3 RXD4 RXD5 RXD6 RXD7 CRS RX_DV RXD0 RXD1 RXD2 RXD3 RXD4 RXD5 RXD6 RXD7 CRS
SMII TX
TX_EN TXD0 TXD1 TXD2 TXD3 TXD4 TXD5 TXD6 TXD7 TX_ER TX_EN TXD0 TXD1 TXD2 TXD3 TXD4 TXD5 TXD6 TXD7 TX_ER
MII RXCLK and TXCLK
MII RXD/TXD
MII TXEN /RXDV

FIG. 2

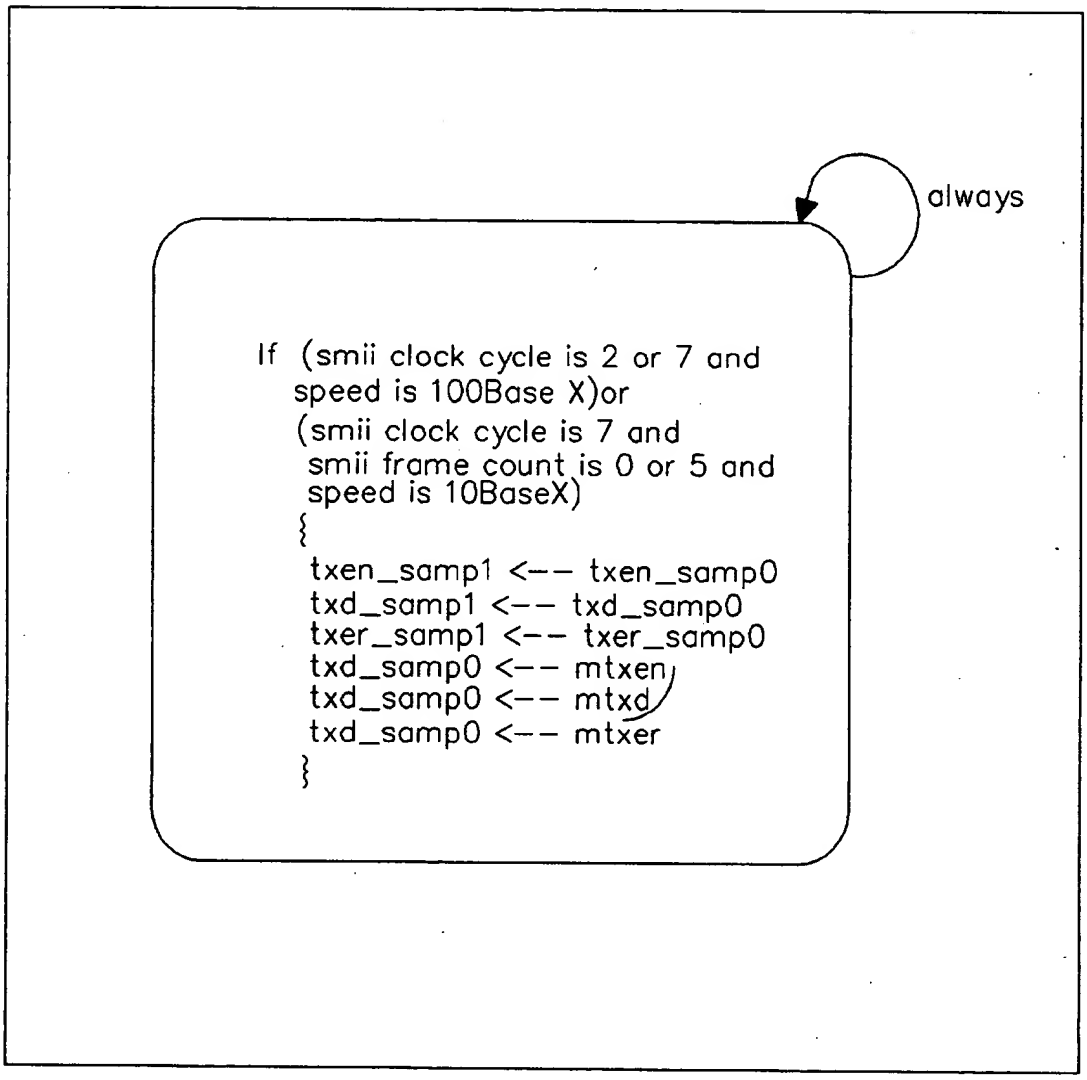
always
If (smii clock cycle is 2 or 7 and
speed is 100Base X)or
(smii clock cycle is 7 and
smii frame count is 0 or 5 and
speed is 10BaseX)
{
txen_samp1 <-- txen_samp0
txd_samp1 <-- txd_samp0
txer_samp1 <-- txer_samp0
txd_samp0 <-- mtxen
txd_samp0 <-- mtxd
txd_samp0 <-- mtxer
}

FIG. 3

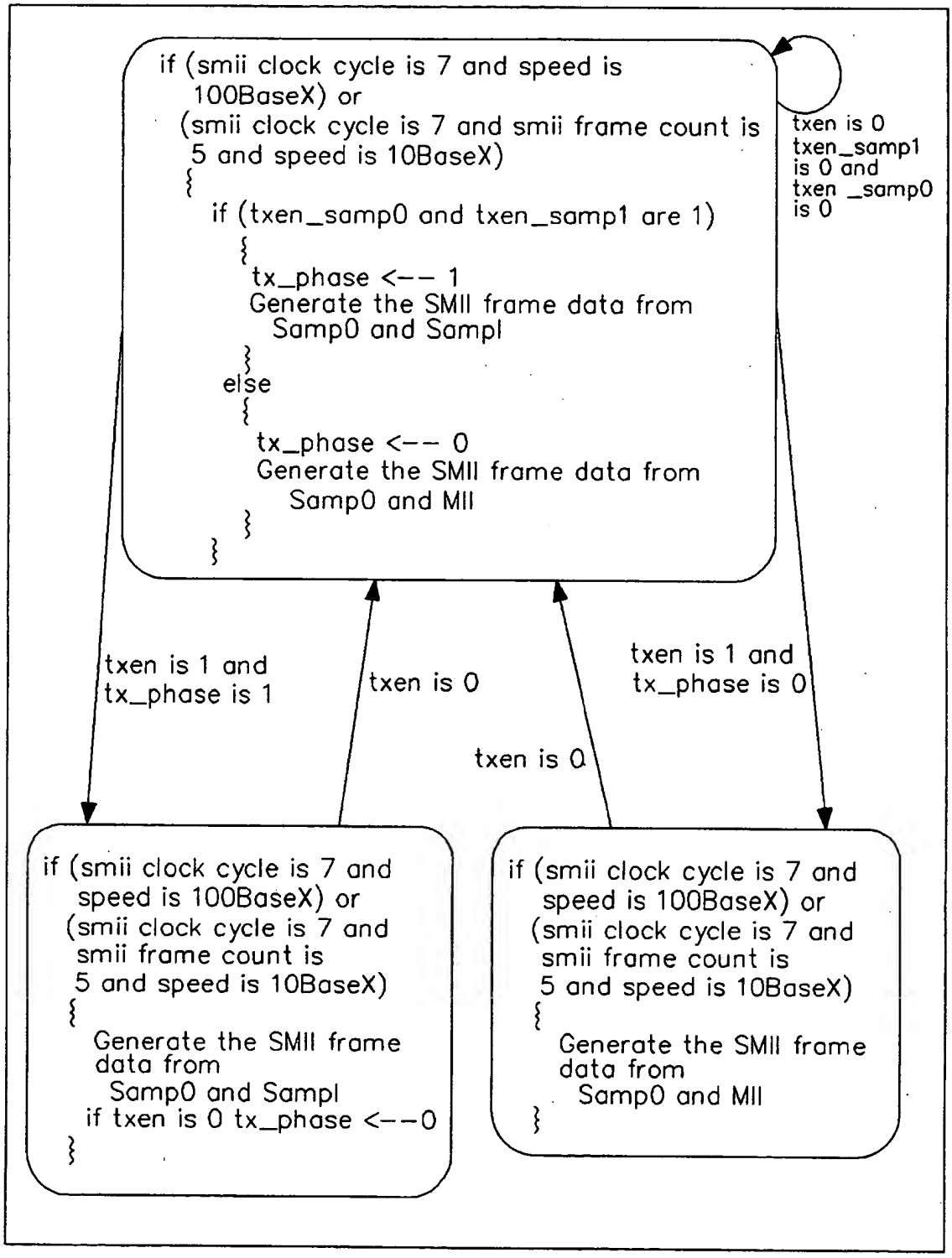
if (smii clock cycle is 7 and speed is 100BaseX) or (smii clock cycle is 7 and smii frame count is 5 and speed is 10BaseX)
{
if (txen_samp0 and txen_samp1 are 1)
{
tx_phase <-- 1
Generate the SMII frame data from Samp0 and Samp1
}
else
{
tx_phase <-- 0
Generate the SMII frame data from Samp0 and MII
}
}
txen is 0 txen_samp1 is 0 and txen _samp0 is 0
txen is 1 and tx_phase is 1
txen is 0
txen is 1 and tx_phase is 0
txen is 0
if (smii clock cycle is 7 and speed is 100BaseX) or (smii clock cycle is 7 and smii frame count is 5 and speed is 10BaseX)
{
Generate the SMII frame data from Samp0 and Samp1
if txen is 0 tx_phase <--0
}
if (smii clock cycle is 7 and speed is 100BaseX) or (smii clock cycle is 7 and smii frame count is 5 and speed is 10BaseX)
{
Generate the SMII frame data from Samp0 and MII
}

FIG. 4

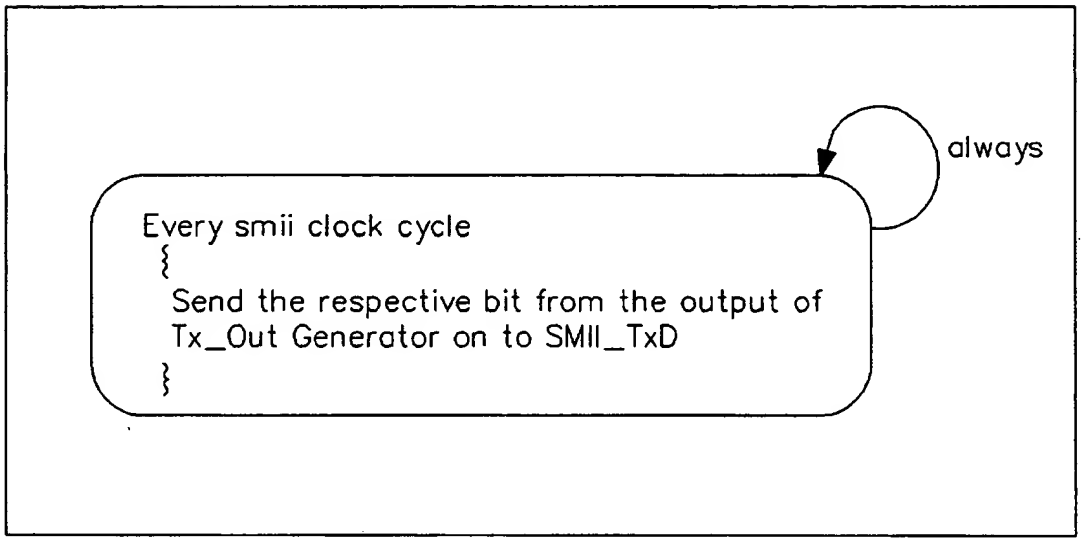

FIG.5

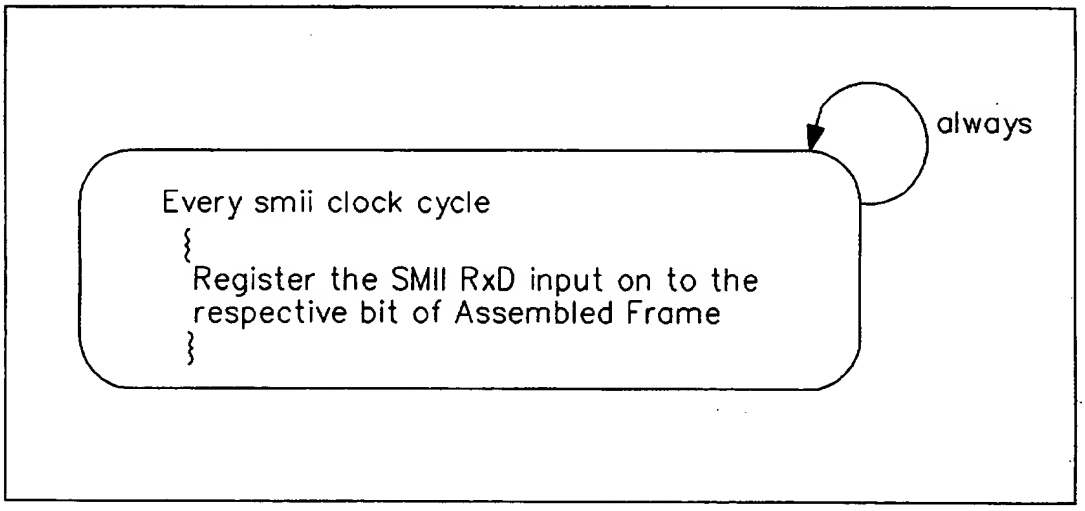

FIG.6

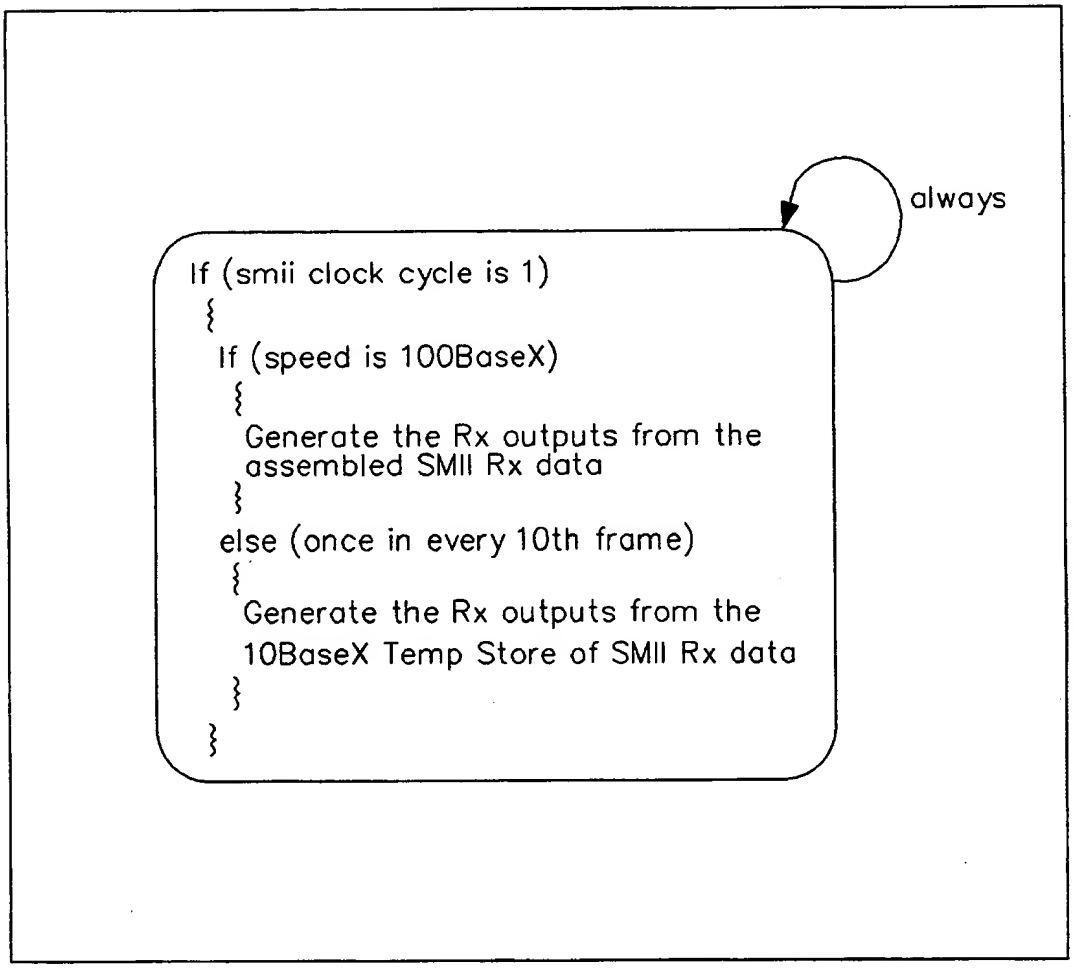
always
If (smii clock cycle is 1)
{
If (speed is 100BaseX)
{
Generate the Rx outputs from the assembled SMII Rx data
}
else (once in every 10th frame)
{
Generate the Rx outputs from the 10BaseX Temp Store of SMII Rx data
}
}

FIG. 7

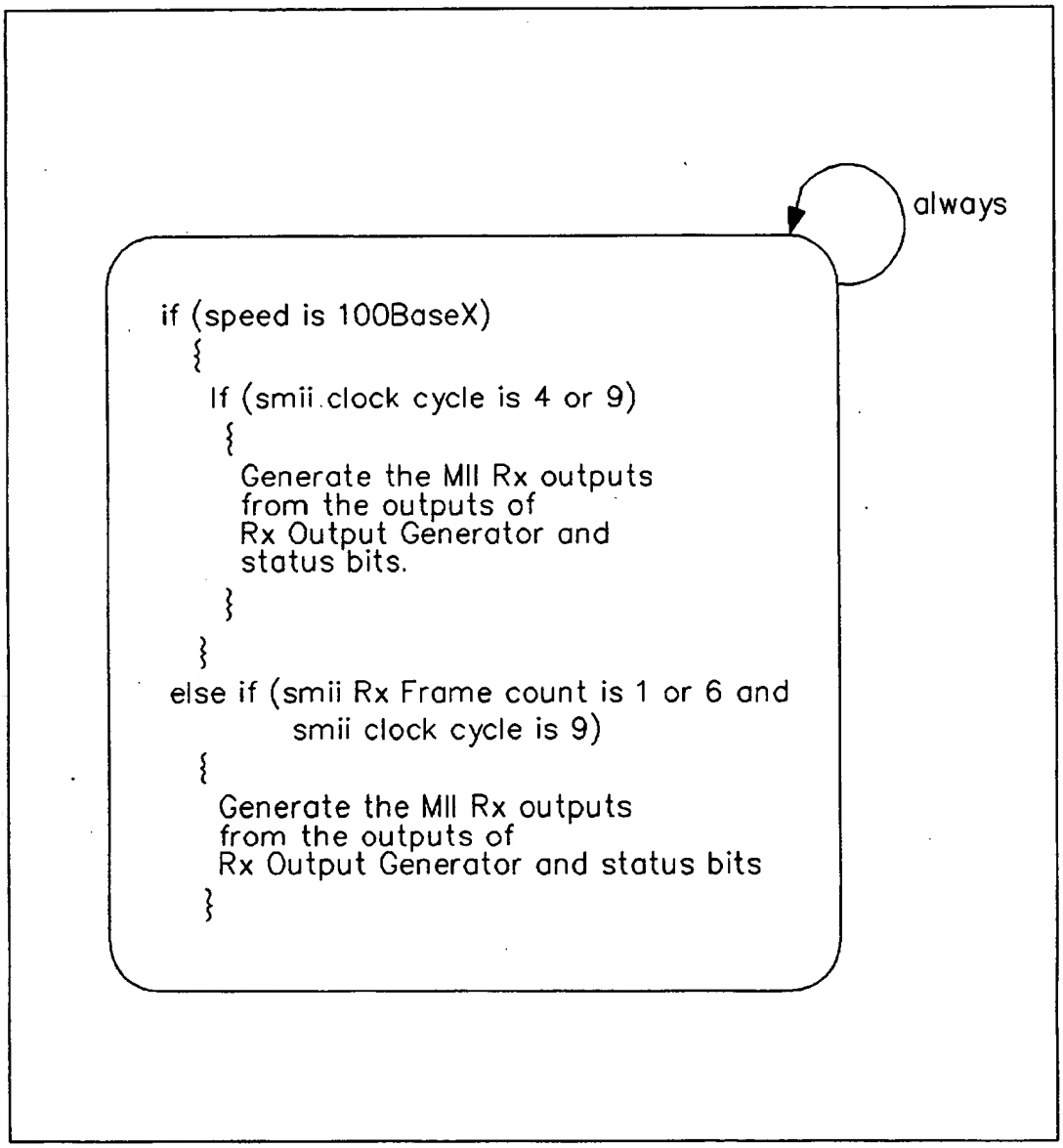

FIG.8